

AMENDMENTS TO THE SPECIFICATION:

Please amend page 1 before line 1 as follows:

OPERATING SYSTEMS

BACKGROUND

1. Technical Field

Please add the following at page 1, between lines 4 and 5:

2. Related Art

Please amend the paragraph beginning at page 1, line 18 as follows:

Applications programs interact with the computers on which they run through operating systems. By using the applications programming interface (API) of the operating system, the applications program can be written in a portable fashion, so that it can execute on different computers with different hardware resources. Additionally, common operating systems such as Linux or Windows provide multi-tasking; in other words, they allow several programs to operate concurrently. To do so, they provide scheduling; in other words, they share the usage of the resources of the computer between the different programs, allocating time to each in accordance with a scheduling algorithm. Operating systems of this kind are very widely used, but they generally make no provision for running real time applications, and they therefore are unsuitable for many control or communications tasks.

For such tasks, therefore, real time operating systems have been developed; one example is ChorusOS (also known as Chorus) and its derivatives. Chorus is available as open source software from: <http://www.experimentalstuff.com/Technologies/ChorusOS/index.html> and Jaluna at <http://www.jaluna.com/>

It is described in “ChorusOS Features and Architecture overview” Francois Armand, Sun

Technical Report, August 2001, 222p, available from

<http://www.jaluna.com/developer/papers/COSDESPERF.pdf> Jaluna.

Please amend the paragraph at page 4, beginning at line 1:

A more similar approach is that of ADEOS (Adaptive Domain Environment for
Operating Systems), described in a White Paper at <http://opersys.com/ftp/pub/Adeos/adeos.pdf>.

ADEOS provides a nanokernel which is intended, amongst other things, for running
multiple operating systems although it appears only to have been implemented with Linux. One
proposed use of ADEOS was to allow ADEOS to distribute interrupts to RTAI (Realtime

Application Interface of Linux) for which see:

<http://www.aero.polimi.it/rtai/applications/>.

Please add the following at page 4, between lines 9 and 10:

BRIEF SUMMARY

Please add the following at page 5, between lines 11 and 12:

BRIEF DESCRIPTION OF THE DRAWINGS

Please add the following at page 6, at line 14:

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS

Please amend the paragraph at page 6, line 18 as follows:

A computer system to which the system is applicable 100 comprises a central processing
unit (CPU) 102, such as a Pentium 4™ CPU available from Intel Corporation, or PowerPC CPU
available from Motorola (the embodiment has been implemented on both), coupled via a system
bus 104 (comprising control, data and address buses) to a read-only memory (ROM) chip 106;

one or more banks of random access memory (RAM) chips (108); disk controller devices 110

(for example ~~IDE or SCSI~~ Integrated Drive Electronics (IDS) or Small Computer System Interface (SCSI) controllers, connected to a floppy disk drive, a hard disk drive, and additional removable media drives such as Digital Video Disk (DVD) drives); one or more input/output ports (112) (for example, one or more Universal Serial Bus (USB) port controllers, and/or parallel port controllers for connection to printer and so on); an expansion bus 14 for bus connection to external or internal peripheral devices (for example the Peripheral component Interconnect (PCI) bus); and other system chips 116 (for example, graphics and sound devices).

Examples of computers of this type are personal computers (PCs) and workstations. However, the application of the invention to other computing devices such as mainframes, embedded microcomputers in control systems, and Personal Digital Assistants (PDAs) (in which case some of the indicated devices such as disk drive controllers may be absent) is also disclosed herein.

Please amend the paragraph at page 8, beginning at line 15:

In use, then, when the computer 100 is started, a bootstrap program stored in Read Only Memory (ROM) 106 accesses the disk controllers 110 to read the file handling part of the operating system from permanent storage on disk into Random Access Memory (RAM) 108, then loads the remainder of the operating system into an area of RAM 108. The operating system then reads any applications from the disk drives via the disk controllers 110, allocates space in RAM 108 for each, and stores each application in its allocated memory space.

Please amend the paragraph at page 13, beginning at line 13:

In this embodiment, the computer 100 was an Intel 386 family processor (e.g. a Pentium processor) and a Motorola PowerPC 750 (Reduced Instruction Set Computer or "RISC")

computer (step 302). The critical operating system 201 (selected at step 304) was the C5 operating system (the real time microkernel of Jaluna-1, an open-source version of the fifth generation of the ChorusOS system, and is available for open source, free download, from <http://www.jaluna.com>).

Please amend the paragraph at page 14, beginning at line 8 as follows:

Trap calls 2012 are added to the critical operating system, to detect states and request some actions in response. A trap call here means a call which causes the processor to save the current context (e.g. state of registers) and load a new context. Thus, where virtual memory addressing is used, the address pointers are changed.

Please amend the paragraph at page 15, beginning at line 4:

Additional “virtual” drivers 2014 are added which, to the operating system, appear to provide access to an input/output (I/O) bus, allowing data to be written to the bus. In fact, the virtual bus driver 2014 uses memory as a communications medium; it exports some private memory (for input data) and imports memory exported by other systems (for output data). In this way, the operating system 201 (or an application running on the operating system) can pass data to another operating system (or application running on it) as if they were two operating systems running on separate machines connected by a real I/O bus.

Please amend the paragraph at page 15, beginning at line 15:

In step 310, the secondary operating system kernel 202 is modified to allow it to function in a multiple operating system environment, which is treated as a new hardware architecture. As in step 306, the boot and initialisationinitialization sequences are modified, to allow the secondary operating system to be started by the hardware resource dispatcher, and to prevent it

from accessing the hardware resources assigned to the other systems, as specified in the hardware resource dispatcher table. As in step 306, trap call 2022 are added, to pass control to the hardware resource dispatcher.

Please amend the paragraph beginning at page 16, line 1:

Native drivers for shared system devices are replaced by new drivers 2028 dealing with devices which have been virtualized by the hardware resource dispatcher (interrupt controller, I/O bus bridges, the system timer and the real time clock). These drivers execute a call to virtual device handlers 416 of the hardware resource dispatcher in order to perform some operations on a respective device of the computer 100. Each such virtual device handler 416 of the hardware resource dispatcher is paired with a “peer” driver routine in the critical operating system, which is arranged to directly interact with the system device. Thus, a call to a virtual device handler is relayed up to a peer driver in the critical system for that virtualized device, in order to make real device access. As in step 306, read and write drivers 2024 for the virtual I/O bus are provided, to allow inter-operating system communications.

Please amend the paragraph at page 16, beginning at line 14:

The interrupt service routines of the secondary operating system are modified, to provide virtual interrupt service routines 2026 each of which responds to a respective virtual interrupt (in the form of a call issued by an interrupt handler routine 412 of the hardware resource dispatcher), and not to respond to real interrupts or events. Routines of the secondary operating system (including interrupt service routines) are also modified to remove masking of hardware interrupts (at least in all except critical operations). In that way, the secondary operating systems 202, ... are therefore pre-emptable by the critical operating system 201; in other words, the secondary

operating system response to a virtual interrupt can itself be interrupted by a real interrupt for the critical operating system 201. This typically includes:

Please amend the paragraph at page 17, beginning at line 13:

To effect this, the Linux kernel ~~207202~~ is modified in this embodiment by adding new virtual hardware resource dispatcher architecture sub trees (nk-i386 and nk-ppc for the I-386 and Power PC variants) with a small number of modified files. Unchanged files are reused in their existing form. The original sub-trees are retained, but not used.

Please amend the paragraph on page 17, beginning at line 21:

- booting and ~~initialising~~initializing itself (402);
- storing a table (403) which stores a list of hardware resources (devices such as ports) and an allocation entry indicating to which operating system each resource is uniquely assigned;
- booting and ~~initialising~~initializing the critical operating system that completes the hardware resource dispatcher allocation tables (404);
- booting and ~~initialising~~initializing secondary operating systems (406)
- switching between operating systems (408);
- scheduling between operating systems (410);
- handling interrupts (using the real time operating system interrupt service routines, and supplying data where necessary to the virtual interrupt service routines of the secondary operating systems) (412);
- handling trap calls from each of the operating systems (414);
- handling access to shared devices from the secondary operating systems (416);

- handling inter-operating system communications on the virtual I/O bus (418).

Please amend the paragraph at page 23, beginning at line 6:

A product which may be supplied in accordance with an aspect of the invention is a development environment product, comprising a computer program which enables the user to select different operating systems to be used, build and select different applications for each operating system, embed the application and operating systems into a deliverable product, and provide for booting of the operating system and launch of executable binaries of the applications.

This is based on, and similar to, the C5 development environment, available from

www.jaluna.com Jaluna.

Please amend the paragraph at page 24, beginning at line 6:

Each operating system kernel then goes through its own initialisation initialization phase (4062, 4064), selecting the resources to be exclusive to that operating system within those remaining in the resource allocation table 404, and starting its initial services and applications.

Please amend the paragraph at page 28, beginning at line 7:

“Fast Error Recovery in CHORUS/OS. The Hot-Restart Technology”. Abrossimov, F. Hermann, J.C. Hugly, et al, Chorus Systems Inc. Technical Report, August 1996, 14p.

available from Jaluna. <http://www.jaluna.com/developer/papers/CSI-TR-96-34.pdf>

Please amend the paragraph at page 28, beginning at line 15:

Having been booted and initialisedinitialized, the real time operating system is running one or more applications 207a, 207b (for example a UDP/IP stack - UDP/IP stands for Universal Datagram Protocol/Internet Protocol) and the secondary operating system is running several applications 208a, 208b (for example a word processor and a spreadsheet). The real time

operating system microkernel 201 and the secondary operating system kernel 202 communicate with the hardware resource dispatcher through the hardware resource dispatcher interface which comprises:

Please amend the paragraph at page 30, beginning at line 14:

Referring to Figure 7, for this example, the system is executing a thread 702 of an application 207a (Figs. 2b and 9a-b) running on the critical operating system 201.

Please amend the paragraph at page 34, beginning at line 19:

Referring to Figure 9b, the changes necessary to migrate this arrangement to one in which the first and second operating systems run on different computers 100, ~~101~~¹⁰⁷ are small; it is merely necessary to change the drivers used by the operating systems, so that they use drivers for a real bus 103 rather than the virtual bus drivers. The system is therefore made more independent of the hardware on which it operates.

Please amend the heading at page 39 before claim 1:

WE CLAIMS
WHAT IS CLAIMED IS